Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2003-39

INTERIOR POINT STABILIZATION FOR COLUMN GENERATION

by

Louis-Martin Rousseau
Michel Gendreau
Dominique Feillet

December 2003

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone : (514) 343-7575 / Télécopieur : (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

104-02001/

PUBLICATION
CRT-2003-39

INTERIOR POINT STABILIZATION FOR COLUMN GENERATION

by

Louis-Martin Rousseau[1]
Michel Gendreau[2]
Dominique Feillet[3]

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1. Département de mathématiques et génie industriel, École Polytechnique de Montréal, and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (louism@crt.umontreal.ca)
2. Département d'informatique et de recherche opérationnelle, Université de Montréal, and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (michelg@crt.umontreal.ca)
3. Laboratoire d'Informatique d'Avignon, 339 chemin des Meinajariés, B.P. 1228, 84311 Avignon, France (dominique.feillet@lia.univ-avignon.fr)

Département d'informatique et de recherche opérationnelle - Publication #1244

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2003

Abstract

Interior Point Stabilization (IPS) is an acceleration method for Column Generation algorithms. Like previous stabilization techniques, it addresses degeneracy problems and convergence difficulties by preventing dual variables from taking extreme values. IPS is however different since it selects a dual solution inside the optimal dual space rather than retrieving an extreme point solution from the LP solver. The general method is presented and applied to the well known case of the Vehicle Routing Problem with Time Windows (VRPTW). Computational results indicate that IPS compares favourably to traditional stabilization approaches, while necessitating fewer parameterizations.

Keywords: column generation, stabilization, vehicle routing with time windows.

Résumé

La stabilisation par point intérieur (Interior Point Stabilization : IPS) est une technique d'accélération pour les méthodes de génération de colonnes. Cette méthode vise à surmonter les problèmes de convergence lente liés à la dégénérescence du problème maître de la décomposition de Dantzig-Wolf. L'IPS est toutefois différente des autres méthodes de stabilisation dans la mesure où elle tente d'identifier un point intérieur de l'espace dual optimal plutôt que de limiter les points possibles retournés par un solveur linéaire. La méthode générale est présentée et appliquée au problème bien connu de confection de tournées de véhicules avec fenêtre de temps. Nos expériences indiquent que l'IPS donne des résultats comparables aux autres méthodes de stabilisation tout en ayant moins de paramètres et en étant plus simplement implanté.

Mots-clés: génération de colonnes, stabilisation, tournées de véhicules avec fenêtres de temps.

Introduction

Column generation was introduced by Dantzig and Wolfe [2] to solve linear programs with decomposable structures. It has been applied to many problems with success and has become a leading optimization technique to solve Routing and Scheduling Problems [4, 1]. However column generation methods often show very slow convergence due to heavy degeneracy problems. This behavior led to the introduction of a stabilization method by du Merle *et al.* [6], which attempted to accelerate convergence by limiting the negative effects of degeneracy. The method proposed here tries to achieve the same goal by using a different approach.

The next section, which gives a brief overview of the column generation framework, is followed by a description of the classical stabilization method. Section 3 introduces Interior Point Stabilization and results are reported in section 4.

1 Column Generation

Column generation is a general framework that can be applied to numerous problems. However, since an example is often useful to give a clear explanation, all results in this paper will be presented with respect to the Vehicle Routing Problem with Time Windows (VRPTW). This is done without loss of generality nor restrictions to deal with this particular problem. The VRP can be described as follows: given a set of customers, a set of vehicles, and a depot, find a set of routes of minimal length, starting and ending at the depot, such that each customer is visited by exactly one vehicle. Each customer having a specific demand, there are usually capacity constraints on the load that can be carried by a vehicle. In addition, there is a maximum amount of time that can be spent on the road. The time window variant of the problem (VRPTW) imposes the additional constraint that each customer must be visited during a specified time interval. One can wait in case of early arrival, but late arrival is not permitted.

In the first application of column generation to the field of Vehicle Routing Problems with Time Windows, presented by Desrochers *et al.* [3], the basic idea was to decompose the problem into sets of customers visited by the same vehicle (routes) and to select the optimal set of routes between all possible ones. Letting r be a feasible route in the original graph (which contains N customers), R be the set of all possible routes r, c_r be the cost of visiting all the customers in r, $A = (a_{ir})$ be a Boolean matrix expressing the presence of a particular customer (denoted by index $i \in \{1..N\}$) in route r, and x_r be a Boolean variable specifying whether the route r is chosen ($x_r = 1$) or not ($x_r = 0$), the Set Partitioning formulation is defined as (S):

$$\min \sum_{r \in R} c_r x_r \tag{1}$$

$$s.t \sum_{r \in R} a_{ir} x_r \quad = 1 \quad \forall i \in \{1..N\} \tag{2}$$

$$x \in \{0,1\}^N \tag{3}$$

This formulation however poses some problems. Firstly, since it is impractical to construct and to store the set R because of its very large size, it is usual to work with a partial set R' that is enriched iteratively by solving a subproblem. Secondly, the Set Partitioning formulation is difficult to solve when R' is small and it allows negative dual values which can be problematic for the subproblem (a negative

dual means there is a negative marginal cost to visit a node...). This is why, in general, the following relaxed Set Covering formulation is used instead as a Master Problem (M):

$$\min \sum_{r \in R'} c_r x_r \tag{4}$$

$$s.t \sum_{r \in R'} a_{ir} x_r \geq 1 \quad \forall i \in \{1..N\} \tag{5}$$

$$x \geq 0 \tag{6}$$

To enrich R', it is necessary to find new routes that offer a better way to visit the customers they contain, that is, routes with a negative reduced cost. The reduced cost of a route is calculated by replacing the cost of an arc (the distance between two customers) d_{ij} by the reduced cost of that arc $c_{ij} = d_{ij} - \lambda_i$, where λ_i is the dual value associated with the covering constraint (5) of customer i. The dual value associated with a customer can be interpreted as the marginal cost of visiting that customer in the current optimal solution (given R'). The objective of the subproblem is then the identification of a negative reduced cost path, i.e., a path for which the sum of the travelled distance is inferior to the sum of the marginal costs (dual values). Such a path represents a new and better way to visit the customers it serves.

The optimal solution of (M) has been identified when there exists no more negative reduced cost path. This solution can however be fractional, since (M) is a relaxation of (S), and thus does not represent the optimal solution to (S) but rather a lower bound on it. If this is the case, it is necessary to start a branching scheme in order to identify an integer solution.

2 Stabilization Problems

Most column generation frameworks depend heavily on marginal costs to guide the search at the subproblem level. In the first iterations of the optimization process, it is possible that the marginal costs associated with each customer are not appropriately estimated by the dual values. For instance, it is possible that in some routes some customers pick up most of the total dual value. If this is the case (as illustrated in figure 1), then in the subproblem a path that visits each of those overweighted customers (B, C) will be considered a good route (with reduced cost of -5), even though it is not the case. With a more realistic distribution of dual values, all nodes would have been given a value of 15 and no more reduced cost path would have been found, thus saving the need for a last iteration.

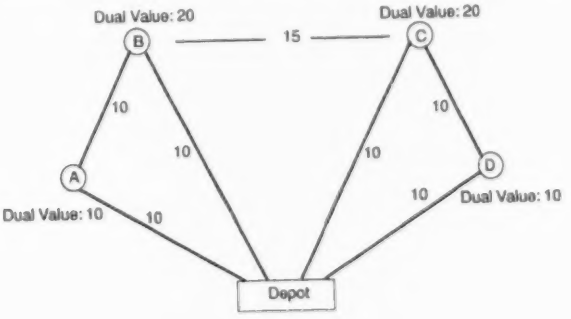

Figure 1: Path $Depot - B - C - Depot$ has a negative reduced cost of -5.

This problem occurs because the Set Covering problem is degenerate and thus its dual has an infinite number of optimal solutions. The standard function that returns dual values in most LP codes usually returns an extreme point of the dual polyhedron. This behavior encourages dual variables to take either very large or very small values. Much better approximations of the real marginal cost would be obtained if the dual variables would take values in the center (or at least the interior) of the optimal dual polyhedron.

To prevent dual variables from taking extreme values, the general strategy proposed in the literature is to try to limit the distance traveled by the dual variables in the dual space from one iteration to another. Two main techniques are available. A first technique consists in defining a box around the previous value of each dual variable and by modifying the Master Problem so that the feasible dual space is limited to the area defined by these boxes. A second technique is to adapt the Master Problem so that any distance separating a dual solution from the previous optimal dual solution is linearly penalized. These two techniques were respectively proposed by Marsten *et al.* [12] and by Kim *et al.* [10] for improving convergence of Kelley's well-known cutting plane algorithm [9].

Du Merle *et al.* [6] have proposed to stabilize column generation with a method that combines these two concepts. They impose soft limits on the dual values λ associated with the constraints (5) of (M). These limits, defined by (d^-, d^+), are first given as parameters to the model and then automatically updated during the resolution. The λ variables can take values outside the given limits (when $\omega^- > 0$ or $\omega^+ > 0$), but the objective is then penalized by $(\varepsilon^-, \varepsilon^+)$. The modified problem (M') now stabilized (and perturbed) is written as:

$$\min \sum_{r \in R'} c_r x_r + \sum_{i \in \{1..N\}} -d_i^- y_i^- + d_i^+ y_i^+ \tag{7}$$

$$s.t \sum_{r \in R'} a_{ir} x_r - y_i^- + y_i^+ \geq 1 \quad \forall i \in \{1..N\} \tag{8}$$

$$y_i^- \leq \varepsilon^- \quad \forall i \in \{1..N\} \tag{9}$$

$$y_i^+ \leq \varepsilon^+ \quad \forall i \in \{1..N\} \tag{10}$$

$$x, y^-, y^+ \geq 0 \tag{11}$$

and the associated dual becomes:

$$\max \sum_{i \in \{1..N\}} \lambda_i - \varepsilon_i^- \omega_i^- - \varepsilon_i^+ \omega_i^+ \tag{12}$$

$$s.t \sum_{i \in \{1..N\}} \lambda_i a_{ir} \leq c_r \quad \forall r \in R' \tag{13}$$

$$\lambda_i + \omega_i^- \geq d_i^- \quad \forall i \in \{1..N\} \tag{14}$$

$$\lambda_i - \omega_i^+ \leq d_i^+ \quad \forall i \in \{1..N\} \tag{15}$$

$$\lambda, \omega^-, \omega^+ \geq 0 \tag{16}$$

This method is subsequently described as Box Stabilization since the bounds (d^-, d^+) on the λ dual variables can be represented by a set of boxes defined by centers (B^c) and sizes (B^s). The mapping between the box representation and the LP formation (12)-(16) is the following: $d_i^- = B_i^c - B_i^s$ and $d_i^+ = B_i^c + B_i^s \; \forall i \in \{1..N\}$. The values of ε^- and ε^+ express the penalty imposed to the objective when the values of λ are chosen outside the box.

In order to use this method efficiently one must define an initializing procedures for B^c and size control procedure for B^s. If we can easily see that initial values of B^c could be provided by any heuristic estimating reasonable marginal costs, there is few information in the literature about the control of the box size and penalty factor (ε) during resolution. The approach described in [5] offers two possible approaches to update the parameters: one is to enlarge the size (B_i^s) of a box when the dual value λ_i (obtained from the LP) lies on or outside its boundary and to reduce it otherwise. The other is to keep increasing the values of ε^- and ε^+ as long as the the subproblem is able to identify new columns and to reduce them otherwise. In both cases when no new column can be found, all boxes are recentered on the last dual values obtained ($B^c = \lambda$). The convergence criterion of the method is that dual λ variables take values which are strictly within $[d^-, d^+]$ or that $\varepsilon^- = \varepsilon^+ = 0$.

Although the approach of du Merle *et al.* [6] is the most widely used for stabilizing column generation, several other stabilization approaches have been proposed in the literature. For complete reviews and discussions on this topic, see Neame [13] and Lübbecke and Desrosiers [11].

3 Interior Point Stabilization

The idea behind Interior Point Stabilization is to generate a dual solution that is an interior point of the optimal dual space rather than an extreme point. The proposed way to achieve this goal is to generate several extreme points of the optimal dual polyhedron and to generate an interior point corresponding to a convex combination of all these extreme points. This section provides details how this is achieved.

3.1 Theoretical issues

As mentioned above, the original relaxed Set Covering problem (M) is written:

$$\min \sum_{r \in R} c_r x_r \tag{17}$$

$$s.t \sum_{r \in R} a_{ir} x_r \geq 1 \quad \forall i \in \{1..N\} \tag{18}$$

$$x \geq 0 \tag{19}$$

The dual of (M) is denoted by (D) and written:

$$\max \sum_{i \in \{1..N\}} \lambda_i \tag{20}$$

$$s.t \sum_{i \in \{1..N\}} \lambda_i a_{ir} \leq c_r \quad \forall r \in R \tag{21}$$

$$\lambda \geq 0 \tag{22}$$

Once (M) is solved to optimality, let R^* be defined as the set of routes r for which $x_r > 0$ (that is some of the columns that are in the basis) and let S be the set of nodes for which the covering constraint (18) is not tight. Using complementary slackness conditions, the optimal dual polyhedron $\mathcal{D}$ containing all optimal values of λ is defined by:

$$\sum_{i\in\{1..N\}} \lambda_i a_{ir} \leq c_r \quad \forall\, r \in R\backslash R^* \tag{23}$$

$$\sum_{i\in\{1..N\}} \lambda_i a_{ir} = c_r \quad \forall\, r \in R^* \tag{24}$$

$$\lambda_i = 0 \quad \forall\, i \in S \tag{25}$$

$$\lambda_i \geq 0 \quad \forall\, i \in \{1..N\}\backslash S \tag{26}$$

To obtain a point in this polyhedron one can simply add a random objective function $u\lambda$ with $u \sim U(0,1)$ and solve the following LP, denoted $(\mathcal{D}^u)$:

$$\max \sum_{i\in\{1..N\}} u_i\lambda_i \tag{27}$$

$$\sum_{i\in\{1..N\}} \lambda_i a_{ir} \leq c_r \quad \forall r \in R\backslash R^* \tag{28}$$

$$\sum_{i\in\{1..N\}} \lambda_i a_{ir} = c_r \quad \forall r \in R^* \tag{29}$$

$$\lambda_i = 0 \quad \forall\, i \in S \tag{30}$$

$$\lambda_i \geq 0 \quad \forall\, i \in \{1..N\}\backslash S \tag{31}$$

If this problem is solved with a simplex based method then the optimal solution of $(\mathcal{D}^u)$ will constitute an extreme point of $\mathcal{D}$. Different instances of $(\mathcal{D}^u)$ can be generated by defining multiple objective functions (u vectors) and thus several extreme points of $\mathcal{D}$ can be obtained. For each u vector generated, both problems $(\mathcal{D}^u)$ and $(\mathcal{D}^{-u})$ are successively solved in order to favor the identification of distant extreme points. Since $\mathcal{D}$ is a convex set, any convex combination of its extreme points will lie within it. Furthermore, except in pathological cases, one would expect convex combinations of several extreme points with strictly positive coefficients to yield an interior point of $\mathcal{D}$. In particular, if we take the average of all obtained extreme points, we should obtain an interior point of $\mathcal{D}$ that is a much better estimate of the real marginal costs.

3.2 Practical issues

This procedure for obtaining an interior point is very simple, but it requires to either write the dual of the master problem or to automatically generate it. This step, which is not very difficult, can be a bit troublesome when additional constraints are present in (M). To avoid it, let us consider the dual of $(\mathcal{D}^u)$ denoted by $(\mathcal{P}^u)$ and defined by:

$$\min \sum_{r\in R} c_r x_r \tag{32}$$

$$s.t \sum_{r\in R} a_{ir} x_r \geq u_i \quad \forall i \in \{1..N\}\backslash S \tag{33}$$

$$\sum_{r\in R} a_{ir} x_r \geq -\infty \quad \forall i \in S \tag{34}$$

$$x_r \geq 0 \quad \forall r \in R\backslash R^* \tag{35}$$

$$x_r \ \ free \quad \forall r \in R^* \tag{36}$$

Interior Point Stabilization can thus be achieved by making simple modifications to the original primal problem. Extreme points of $\mathcal{D}$ are obtained if, after modifying the right-hand-side, $(\mathcal{P}^u)$ is solved and its dual values are collected. The interior point is again generated by collecting a number of extreme points (here multipliers of $(\mathcal{P}^u)$) and averaging them.

Furthermore, it is not really necessary to modify all right-hand-side terms (those set to random values). In order to improve efficiency, only a small subset of these terms can be changed so that the LP solver is able to resolve the new problem efficiently.

This procedure is very general and could turn out to be very useful in numerous column generation applications. The extra time needed to compute the solutions of $(\mathcal{P}^u)$ is sometime negligible compared to the time necessary to solve one subproblem; and as we will see in the results section this procedure can significantly cut down the number of iterations needed to reach optimality.

3.3 Implications for the Subproblem

In the context of unstabilized column generation, the first iterations are usually characterized by extreme dual values (either 0 or large numbers). At the subproblem level, this tends to make the search for negative reduced cost paths much easier since all nodes which were given a 0 marginal cost can be discarded. Of course, the paths generated are of poor quality and these first iterations are only useful in stabilizing the whole process.

Unfortunately, when one introduces a stabilization technique one also makes the first few subproblems harder. In some cases this can be problematic, namely when the subproblem solution algorithm tries to identify all non dominated paths of negative reduced cost. This is due to the improved estimation of the marginal costs that yields more average values (fewer 0) and thus many more negative paths. The performance decrease does not lie in the identification of these paths but rather in the fact that many more paths can be identified.

Since there is no real need to generate all negative reduced cost paths, it is possible, and sometimes essential, to limit the total number of columns generated. This can be done easily (and is probably already implemented) in almost all subproblem algorithms whether they are based on Dynamic Programming ([3],[4],[1]) or on Constraint Programming ([8], [14]). A simple implementation is to stop the subproblem when a sufficient number of paths have been identified, but more sophisticated strategies could be devised.

4 Experimental Results

In this section we attempt to demonstrate the effectiveness of Interior Point Stabilization with regard to two criteria. We first evaluate IPS as a simple (almost parameter free) stabilization method for column generation and then compare it to traditional Box Stabilization. The problem chosen to perform experiments is the Vehicle Routing Problem with Time Windows (VRPTW).

4.1 Benchmarks

We have tested the proposed method on the well-known Solomon instances [15]. The geographical data are randomly generated in problem sets R1, clustered in problem sets C1, and a mix of random and clustered structures in problem sets RC1. The customer coordinates are identical for all instances within one type (i.e., R, C and RC). The instances differ with respect to the width of the time windows.

Some have very tight time windows, while others have time windows which are hardly constraining. Each instances contains 100 customers.

In the data sets, the distance matrix is not explicitly stated, but customer locations are given. Euclidean distance between these customers are calculated with one decimal point, to allow comparison with other published methods.

4.2 Implementation of Column Generation

The column generation framework used to evaluate the stabilization method is quite straightforward since the Restricted Master Problem is exactly the one described by (4)-(6). Columns are generated using a resource constrained elementary shortest path algorithm based on dynamic programming (Feillet *et al.* [7]). At each iteration of the subproblem, the number of non-dominated columns generated at the destination node is limited to 500. The maximum allowed time to find a solution was set to 3600 seconds.

As the aim of this project was to assess the impact of stabilization, no particular effort was made to optimize the implementation of the framework. Furthermore, since a reduction of both the number of iterations and the CPU time can be observed at the root node of the search tree, we do not include in the following experiments the Branch and Price search for integer solutions.

Computational experiments were carried out on a 1.8 GHz Pentium 4 PC with 256 Megabytes of RAM. The Master Problem was modeled and solved with Cplex 7.5 leaving all parameters to their default values.

4.3 Implementation of Stabilization Methods

4.3.1 Box Stabilization

In order to be able to perform reliable comparisons with IPS, we re-implemented the Box Stabilization method described in section (2) using two different sets of initial values. In the first implementation, potentially good initial values are defined using some knowledge we have about the problem at hand. In the present case, we estimated the marginal costs of each node as the cost of its cheapest outgoing arc ($\lambda_i = min_{j \in \{1..N\}} c_{ij}$) and initialized each box center with these values ($B_i^c = \lambda_i, \forall i \in \{1..N\}$).

The second strategy is to use as initial centers for the stabilizing boxes the optimal dual value associated with each node. Each problem is solved a first time and the optimal dual values (λ_i^*) are stored and used as initial box centers ($B_i^c = \lambda_i^*, \forall i \in \{1..N\}$) for the stabilized resolution. Clearly this process is not applicable in practice since each problem needs to be solved twice. However, since it is not clear a priori how Box Stabilization is sensitive to the quality of the initial boxes, this last evaluation permits to determine the potential best gain achievable with this technique.

For both strategies, ε^- and ε^+ are initially set to 0.001 and the parameter control process we used is the following: all boxes are initialized to a size of 0.2 ($B_i^s = 0.1, \forall i \in \{1..N\}$). If the subproblem is able to find negative reduced cost path, then the values of ε^- and ε^+ are increased by 10%. However, when there is no more such column, the values of ε^- and ε^+ are considerably decreased (divided by 1000) and every box is recentered on the last dual value ($B_i^c = \lambda_i, \forall i \in \{1..N\}$). This is performed in order to achieve convergence through the second criterion ($\varepsilon^- = \varepsilon^+ = 0$). We have experimented with different updating strategies and the one exposed here provided the best results.

4.3.2 Interior Point Stabilization

IPS was implemented using the model (P) (32)-(36) described in section 3.2. The only parameter to set is the number of sample points to identify in order to calculate an interior point of the dual optimal space. The tradeoff is that a larger number of points will produce a more accurate approximation but also requires more computing time (since a LP has to be solved for each point). Preliminary experiments tend to show that a *sufficient* number of points is enough to ensure stabilization and that computing more points does not really improve the global performance of the column generation framework. For the set of VRPTW instances at hand, 20 seemed to be a sufficient number of sample points.

In the present case, since solving (P) takes less then 0.01 second, we were able the modify the right-hand-side of all constraints (33) without significantly slowing down the global resolution process.

4.4 Comparisons and Discussion

Table 1 compares the effectiveness of the different stabilization approaches. For each algorithm, it indicates the computing time (*time*) and the number of iterations (*iter*). The optimal solution value is identical for all approaches and is mentioned in column *value*.

Unstabilized corresponds to the standard column generation scheme for the solution of the Master Problem (4)-(6). *BoxHeur* and *BoxOpt* stand for the Box stabilization method, respectively with an heuristic or an optimal choice for the box centers. *IPS* is for Interior Point Stabilization.

		Unstabilized		BoxHeur		BoxOpt		IPS	
Instances	Value	Time	Iter	Time	Iter	Time	Iter	Time	Iter
r101	1631.15	4.35	33	5.19	38	4.28	29	5.33	23
r102	1466.60	38.77	43	45.83	48	27.77	32	26.86	25
r103	1206.78	307.89	68	318.49	59	235.91	44	226.06	42
r105	1346.14	17.04	46	17.88	46	16.00	40	17.17	34
r106	1226.91	181.85	69	192.03	56	172.48	49	147.73	44
r107	1053.60	3451.77	88	3569.92	74	3018.99	66	2656.92	58
r109	1134.28	81.93	62	71.65	50	69.68	45	65.16	40
r110	1055.57	510.29	77	506.29	58	447.92	45	413.44	45
r111	1034.73	817.61	88	864.52	69	656.37	55	690.78	56
c101	827.30	19.25	87	12.17	45	4.18	16	7.56	23
c102	827.30	844.45	110	747.62	88	554.87	38	230.69	38
c105	827.30	30.01	81	28.27	70	8.47	21	13.05	27
c106	827.30	46.35	84	37.50	62	14.92	25	18.50	24
c107	827.30	32.27	74	38.31	65	17.49	30	17.85	30
c108	827.30	123.91	118	84.29	78	26.22	25	43.52	37
c109	827.30	317.53	124	296.89	101	67.45	22	80.97	27
rc101	1584.09	11.19	39	11.69	39	10.85	38	13.40	33
rc102	1406.26	81.63	56	85.63	54	69.91	43	68.48	38
rc103	1225.65	916.65	76	978.19	74	768.04	54	802.75	52
rc105	1471.93	41.84	51	48.00	50	38.10	38	36.75	33
rc106	1318.80	66.03	56	76.44	57	54.55	38	62.89	41
rc107	1183.37	514.70	68	524.21	61	498.82	48	469.34	47

Table 1: Comparison of IPS with box-stabilized and unstabilized column generation algorithms

Results reported in table 1 are the time and number of iterations needed to identify the optimal solution to the linear relaxation of (1)-(3). Results show a clear improvement when implementing the

	BoxHeur		BoxOpt		IPS	
Class	Time	Iter	Time	Iter	Time	Iter
R	103%	87%	86%	71%	79%	64%
C	88%	75%	49%	26%	29%	30%
RC	106%	97%	88%	75%	89%	71%

Table 2: Average reduction of time and iterations of stabilization methods over the unstabilized algorithm

Interior Point Stabilization within the standard column generation scheme, both in time and number of iterations.

As shown in table 2, stabilization appears particularly effective for C instances, where customers are clustered. This could be easily interpreted: in these instances clusters are commonly serviced with a single vehicle, which should discard every column not visiting all nodes in a cluster. The visit of every customers in a cluster is enhanced by the IPS, since low values for dual variables (and notably null dual variable) are avoided. Improvements are however smaller for the R and RC instances.

Experiments were also conducted on IPS using only one extreme point to artificially simulate an unstabilized behavior. The results are obtained by this method were significantly poorer then those obtained by simply retrieving the marginal cost from the LP solver (here ILOG Cplex 7.5). This indicates that the LP code has some sort of stabilizing mechanism, probably related to its incremental solving capabilities.

The table also compares the different stabilization approaches. IPS performs better than the BoxHeur algorithm and exhibits results similar to BoxOpt. These results tend to demonstrate that even a very clever implementation of the Box Stabilization would not be far better than IPS. On the contrary, IPS outperforms a standard implementation of BoxStep, though much simpler and intuitive.

5 Conclusion

Interior Point Stabilization, a new stabilization method, was introduced and tested on the Vehicle Routing Problem with Time Windows. This method, which is based on a different marginal cost estimator (an interior point of the dual optimal space), can significantly improve the behavior of column generation frameworks. Furthermore IPS is intuitive, simple to implement and almost parameter free.

Future work on this technique involves its application to different applications in order to clearly evaluate in which contexts it can be useful.

Acknowledgment

Financial support for this research was provided by the Natural Science and Engineering Research Council of Canada.

References

[1] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-Price: Column Generation for Huge Integer Programs. *Operations Research*, 46:316–329, 1998.

[2] G.B. Dantzig and P. Wolfe. Decomposition principles for linear programs. *Operations Research*, 8:101–111, 1960.

[3] M. Desrochers, J. Desrosiers, and M.M. Solomon. A New Optimisation Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research*, 40:342–354, 1992.

[4] J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. Time Constrained Routing and Scheduling. In M.O. Ball, T.L. Magnanti, C.L. Monma, and Nemhauser G.L., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–139. North-Holland, Amsterdam, 1995.

[5] O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilisation dans le cadre de la génération de colonnes. Publication G-97-08, GERAD, 1997.

[6] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized Column Generation. *Discrete Mathematics*, 194:229–237, 1999.

[7] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Technical report, Laboratoire Informatique d'Avignon, 2003.

[8] U. Junker, S.E. Karisch, N. Kohl, B. Vaaben, T. Fahle, and M. Sellmann. A Framework for Constraint Programming Based Column Generation. In *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, pages 261–274, 1999.

[9] J.E. Kelley. The Cutting Plane Method for solving Convex Programs. *Journal of the SIAM*, 8:703–712, 1960.

[10] S. Kim, K.-N. Chang, and J.-Y. Lee. A descent method with linear programming subproblems for nondifferentiable convex optimization. *Mathematical Programming*, 71:17–28, 1995.

[11] M.E. Lübbecke and J. Desrosiers. Selected topics in Column Generation. Technical Report G-2002-64, GERAD, Montreal, December 2002.

[12] R.E. Marsten, W.W. Hogan, and J.W. Blankenship. The BOXSTEP Method for Large-Scale Optimization. *Operations Research*, 23:389–405, 1975.

[13] P.J. Neame. *Nonsmooth Dual Methods in Integer Programming.* PhD thesis, University of Melbourne, 1999.

[14] L.-M. Rousseau, M. Gendreau, and G. Pesant. Solving Small VRPTWs with Constraint Programming Based Column Generation. In *Proceedings of the Forth Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 2002.

[15] M. M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research*, 35:254–265, 1987.